**WHAT IS CLAIMED IS:**

1. A processor for determining a minimum value of a plurality of values stored in source registers and determining an index value of a source register having the minimum value, the processor comprising:
a destination register;
a first source register storing a first value;
a second source register storing a second value;
means for comparing the first value stored in the first source register with the second value stored in the second source register;
means for storing the first value in the destination register when the first value is less than or equal to the second value; and
means for concatenating the index value with the second value into a concatenated value and storing the concatenated value in the destination register when the second value is less than the first value.

2. The processor of Claim 1, wherein the means for comparing, the means for storing and the means for concatenating are adapted to execute sequentially within one processor cycle.

3. The processor of Claim 1, wherein the first source register and the destination register comprise a same register.

4. The processor of Claim 1, wherein the second source register and the destination register comprise a same register.

5. The processor of Claim 1, wherein the first value is stored in N low-order bits of the first source register and the second value is stored in N low-order bits of the second register, N being an integer value.

6. The processor of Claim 1, wherein the first source register and the second source register each include an active status bit to indicate a status of the respective register, and wherein a value of a register having an active status is less than a value of a register having an inactive status.

7. A processor for determining a minimum value of a plurality of values stored in source registers and determining an index value of source register having the minimum value, the processor comprising:

means for determining a first minimum value of a first value and a second value;

means for determining a second minimum value of a third value and a fourth value;

means for storing the first minimum value in a first portion of a first destination register and the second minimum value in a second portion of the first destination register; and

means for storing a first index value associated with the first minimum value in a first portion of a second destination register and a second index value associated with the second minimum value in a second portion of the second destination register;

wherein the means for determining the first minimum value and the means for determining the second minimum value are adapted to execute in parallel.

8. The processor of Claim 7, wherein the means for determining and the means for storing are adapted to execute sequentially within one processor cycle.

9. The processor of Claim 7, wherein each of the first, second, third, and fourth values includes an active status bit to indicate a status, and wherein a value having an active status is less than a value having an inactive status.

10. The processor of Claim 7, wherein a value having "11" as its two most significant bits is less than a value having "00" as its two most significant bits.

11. The processor of Claim 7, wherein the first source register and the destination register comprise a same register.

12. The processor of Claim 7, wherein the second source register and the destination register comprise a same register.

13. A method for determining a minimum value and a corresponding index value of a plurality of source registers of a processor, the method comprising the steps of:
for each of the plurality of source registers, comparing a value stored in the source register with a value stored in a destination register;
concatenating the value stored in the source register with an index value associated with the source register and storing the concatenated value in the destination register when the value stored in the source register is less than the value stored in the destination register; and
wherein the destination register initially includes an index value and a value of a first source register of the plurality of source registers.

14. The method of Claim 13, wherein the steps of comparing, concatenating, and storing are implemented by a single processor instruction.

15. The method of Claim 13, wherein the processor instruction is executed within one processor cycle.

16. The method of Claim 13, wherein each of the plurality of values represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing.

17. A customer premise equipment (CPE) comprising:
a network interface operably connected to a first network segment;
a network interface operably connected to a second network segment; and
a processor operably connected to the network interfaces and being adapted to:

compare a first value stored in a first source register of the processor with a second value stored in a second source register of the processor;

store the first value in a first destination register of the processor when the first value is less than or equal to the second value; and

store the second value in the first destination register of the processor and an index value in a second destination register of the processor when the second value is less than the first value, the index value representing the second source register.

18. The CPE of Claim 17, wherein the processor is further adapted to compare the first and second values and store values in the destination registers within one processor cycle.

19. The CPE of Claim 17, wherein the first value and second value each represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing for at least one data stream between the first network segment and the second network segment.

20. A processor for determining a maximum value of a plurality of values stored in source registers and determining an index value of a source register having the maximum value, the processor comprising:

a destination register;

a first source register storing a first value;

a second source register storing a second value;

means for comparing the first value stored in the first source register with the second value stored in the second source register;

means for storing the first value in the destination register when the first value is greater than or equal to the second value; and

means for concatenating the index value with the second value into a concatenated value and storing the concatenated value in the destination register when the second value is greater than the first value.

21. The processor of Claim 20, wherein the means for comparing, the means for storing and the means for concatenating are adapted to execute sequentially within one processor cycle.

22. The processor of Claim 20, wherein the first source register and the destination register comprise a same register.

23. The processor of Claim 20, wherein the second source register and the destination register comprise a same register.

24. The processor of Claim 20, wherein the first value is stored in N low-order bits of the first source register and the second value is stored in N low-order bits of the second register, N being an integer value.

25. The processor of Claim 20, wherein the first source register and the second source register each include an active status bit to indicate a status of the respective register, and wherein a value of a register having an active status is greater than a value of a register having an inactive status.

26. A processor for determining a maximum value of a plurality of values stored in source registers and determining an index value of source register having the maximum value, the processor comprising:

means for determining a first maximum value of a first value and a second value;

means for determining a second maximum value of a third value and a fourth value;

means for storing the first maximum value in a first portion of a first destination register and the second maximum value in a second portion of the first destination register; and

means for storing a first index value associated with the first maximum value in a first portion of a second destination register and a second index value associated with the second maximum value in a second portion of the second destination register;

wherein the means for determining the first maximum value and the means for determining the second maximum value are adapted to execute in parallel.

27. The processor of Claim 26, wherein the means for determining and the means for storing are adapted to execute sequentially within one processor cycle.

28. The processor of Claim 26, wherein each of the first, second, third, and fourth values includes an active status bit to indicate a status, and wherein a value having an active status is greater than a value having an inactive status.

29. The processor of Claim 26, wherein a value having "11" as its two most significant bits is greater than a value having "00" as its two most significant bits.

30. The processor of Claim 26, wherein the first source register and the destination register comprise a same register.

31. The processor of Claim 26, wherein the second source register and the destination register comprise a same register.

32. A method for determining a maximum value and a corresponding index value of a plurality of source registers of a processor, the method comprising the steps of:

for each of the plurality of source registers, comparing a value stored in the source register with a value stored in a destination register;

concatenating the value stored in the source register with an index value associated with the source register and storing the concatenated value in the destination register when the value stored in the source register is greater than the value stored in the destination register; and

wherein the destination register initially includes an index value and a value of a first source register of the plurality of source registers.

33. The method of Claim 32, wherein the steps of comparing, concatenating, and storing are implemented by a single processor instruction.

34. The method of Claim 32, wherein the processor instruction is executed within one processor cycle.

35. The method of Claim 32, wherein each of the plurality of values represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing.

36. A customer premise equipment (CPE) comprising:
a network interface operably connected to a first network segment;
a network interface operably connected to a second network segment; and
a processor operably connected to the network interfaces and being adapted to:
compare a first value stored in a first source register of the processor with a second value stored in a second source register of the processor;
store the first value in a first destination register of the processor when the first value is greater than or equal to the second value; and
store the second value in the first destination register of the processor and an index value in a second destination register of the processor when the second value is greater than the first value, the index value representing the second source register.

37. The CPE of Claim 36, wherein the processor is further adapted to compare the first and second values and store values in the destination registers within one processor cycle.

38. The CPE of Claim 36, wherein the first value and second value each represents a due timestamp of a corresponding input queue for implementing Weighted Fair Queuing for at least one data stream between the first network segment and the second network segment.